IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

SECURE AUTHENTICATED NETWORK CONNECTIONS

Inventor(s):

Gary Kiwimagi

Charles McJilton

ATTORNEY'S DOCKET NO. CN1-019US

SECURE AUTHENTICATED NETWORK CONNECTIONS

PRIORITY CLAIM

[0001] This application is a continuation-in-part of co-owned U.S. Patent Application Ser. No. 10/726,231 for "Secure Network Connections" of Kiwimagi, et al. (Attorney Docket No. CN1-015US), filed December 1, 2003, hereby incorporated herein for all that it discloses.

TECHNICAL FIELD

[0002] The described subject matter relates to networks for electronic computing, and more particularly to systems and methods of establishing secure authenticated network connections for electronic computing systems.

BACKGROUND

[0003] The ability to automatically control one or more functions in a building (e.g., lighting, heating, air conditioning, security systems) is known as building automation. Building automation systems may be used, for example, to automatically operate various lighting schemes in a house. Of course building automation systems may be used to control any of a wide variety of other functions, more or less elaborate than controlling lighting schemes.

[0004] It is often desirable to remotely access the building automation system to monitor and/or change various functions of the building automation system. For

example, a homeowner planning to return home from a vacation earlier than initially expected may want to change the building automation system from a vacation mode to an "every-day" mode prior to the occupants returning home. In another example, an integrator may be responsible for installing and/or maintaining automation systems for a number of customers and may want to remotely access a customer's automation system to assist the customer. These examples are merely illustrations of two types of remote access that may be desired as there are others too numerous to discuss.

[0005] Building automation systems may be remotely accessed via networks such as the Internet or telephone networks. However, providing remote access over a public communication network also makes the building automation system vulnerable to unauthorized access, e.g., by hackers. It is therefore desirable to provide remote access via a secure authenticated connection.

SUMMARY

[0006] Implementations described and claimed herein provide access, e.g., to building automation systems among other electronic computer systems, via a secure authenticated network connection. A secure authenticated network connection may be established in a network environment according to one implementation between a client and a system node (e.g., a server controlling the building automation system). The system node provides its network address to a

control node. When the client desires access to the system node, the client requests the network address from the control node. The control node authenticates the client as an authorized user. If the client is an authorized user, the control node provides session information to the system node, the client, and a data node. The client and the system node then use the session information to request access to each other via the data node.

[0007] In some implementations, articles of manufacture are provided as computer program products. One implementation of a computer program product provides a computer program storage medium readable by a computer system and encoding a computer program for establishing a secure authenticated connection. Another implementation of a computer program product may be provided in a computer data signal embodied in a carrier wave by a computing system and encoding the computer program to establish a secure authenticated network connection.

[0008] The computer program product encodes a computer program for executing on a computer system a computer process that generates session information at the control node for a client, a system node, and the data node if the client and the system node satisfy at least one condition for accessing each other. The data node receives a request from the client to access the system node and a request from the system node to access the client, and then establishes a secure

connection between the client and the system node based at least in part on the session information.

[0009] In another exemplary implementation, a method is provided. The method may be implemented to generate session information for a client, a system node, and a data node if the client and the system node satisfy at least one condition for accessing each other. The data node receives a request from the client to access the system node and a request from the system node to access the client. A secure authenticated connection is established between the client and the system node via the data node based at least in part on the session information.

[0010] In yet another exemplary implementation, a system is provided for establishing a secure authenticated network connection between a client and a system node. The system comprises a control node linked to the client and the system node, the control node providing the client and the system node with session information if the client and the system node are authorized to access each other. A data node is communicatively coupled to the control node. The data node receives the session information from the control node and establishes a secure authenticated connection between the client and the system node via the data node based at least in part on the session information.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Fig. 1 is a schematic illustration of an exemplary network for establishing a secure authenticated connection;

[0012] Fig. 2 is a schematic illustration showing an exemplary implementation of electronic computing systems that can be used to establish a secure authenticated connection over a network;

[0013] Figs. 3(a) through (f) illustrate exemplary operations to establish a secure authenticated connection over a network; and

[0014] Fig. 4 is a schematic illustration of an exemplary computing device that can be utilized to establish a secure authenticated network connection.

DETAILED DESCRIPTION

[0015] A user may desire to connect to a building automation system to access various automation functions (e.g., lighting, security, and climate controls) for the building. Configuration/monitoring software (e.g., a web-enabled application) may be provided via a server computer so that the user can use any available computer with a network connection. Alternatively, the integrator's laptop may have the configuration/monitoring software installed.

[0016] In one example, a homeowner may visit an Internet café while on vacation and access his or her home automation system to monitor security or

adjust the thermostat prior to returning home. In another example, an integrator may use a desktop or laptop computer to access a customer's automation system to assist the customer with an automation function (e.g., to change a lighting or climate control scheme). Of course remote access to the building automation system may be desired for any of a wide variety of other reasons as well.

[0017] Access to the building automation system is preferably established via a secure authenticated network connection. Briefly, a secure authenticated network connection may be established in a network environment between a client, such as the integrator's laptop PC, and a system node provided with the building automation system.

[0018] Although exemplary implementations are described herein with reference to building automation systems, it should be understood that the scope is not limited to use with building automation systems and the invention may also find application in a number of different types of electronic computing systems now known or later developed.

Exemplary Architecture

[0019] Fig. 1 is a schematic illustration of an exemplary networked computing system 100 in which a secure authenticated network connection may be established according to one implementation. The networked computer system 100 may include one or more communication networks 110, such as a local area network

(LAN) and/or wide area network (WAN). A control node 120 and data node 125 may be provided to facilitate a secure authenticated connection between one or more clients 130a, 130b, 130c (hereinafter, generally referred to as 130) and a system node 140 (e.g., a server computer implemented in a building automation system at building 145).

[0020] As used herein, the term "node" is used to refer to hardware and software (entire computer system) used to perform various network services. A node may include one or more computing systems, such as a server, that also runs other applications or that is dedicated only to server applications. A node connects to a network via a communication connection, such as a dial-up, cable, or DSL connection via an Internet service provider (ISP).

[0021] A node may provide services to other computing or data processing systems or devices. For example, system node 140 may be implemented as a server computer to start processes in a building automation system. System node 140 may also provide other services, such as Internet and email services. Control node 120 and data node 125 may also be implemented as one or more server computers to broker security and optionally provide application software to the client, as will be discussed in more detail below.

[0022] As used herein, the term "client" refers to the hardware and software (the entire computer system) used to perform various computing services. A client may include a computing system(s), such as a stand-alone personal desktop or

laptop computer (PC), workstation, personal digital assistant (PDA), or appliance, to name only a few. A client also connects to a network via a communication connection, such as a dial-up, cable, or DSL connection via an Internet service provider (ISP) or may connect directly into a LAN, e.g., for the building automation system via network connection.

[0023] Fig. 2 is a schematic illustration showing an exemplary implementation of computer systems that can be connected on a network 200. According to this implementation, a control node 210 and a data node 215 may cooperate to establish a secure authenticated connection (e.g., via network 200) between a client 220 and a system node 230.

[0024] System node 230 may be implemented, e.g., as a server computer operating a building automation system. System node 230 may include application software (not shown). For example, application software may be provided to monitor the status of the building automation system, and administer various automation functions. System node may also serve as a central repository for program code that controls the various building automation devices. Client 220 may access system node 230 to control, configure, and/or monitor the system node 230 (e.g., building automation system).

[0025] System node 230 is identified on the network by a network address 235. The network address may be any address that identifies a system node 230 on a network 200. By way of example, the network address may include an Internet

Protocol (IP) address, although higher level addresses (e.g., a domain name) may also be used in other implementations. System node 230 provides its network address 235 to the control node 210 during a registration operation so that the system node 230 can be identified on the network, e.g., by the client 220.

[0026] The network address may be a dynamic (i.e., changing) network address. Use of a dynamic network address adds another layer of security to the network connection because a client 220 cannot simply store the network address and reuse it at a later time to regain access to the system node 230. Instead, the dynamic network address is updated at the control node 210 and the client 220 has to request the current network address from the control node 210 before the client 220 is able to access the system node 230.

[0027] Client 220 may be implemented in a laptop or desktop computer, or in any other suitable device which is capable of establishing a network connection, and sending and/or receiving data over that network connection (e.g., a PDA or mobile phone). Client 220 may include security credentials 225 (e.g., UserID and password) that may be presented to the control node 210 and/or the data node 215 to authenticate the client 220 for access to the system node 230.

[0028] Client 220 may also include a user interface module 226. User interface module 226 may be implemented as program code (e.g., software). User interface module 226 may be used, for example, by a homeowner, integrator, or other user to send and receive messages or process transactions.

[0029] Client 220 may request access to the system node 230 (i.e., a client session) by control node 210. In an exemplary implementation, control node 210 includes an authorization module 211. Authorization module 211 may be implemented as computer readable program code (e.g., software, firmware) stored in computer readable storage or memory and executable by a processor (or processing units) operatively associated with the control node 210. Authorization module 211 performs operations, such as authorizing the client (e.g., based on security credentials 225) and generating session information in response to a request by a client 220 to access a system node.

[0030] Session information may include data in any suitable format to identify a client session to the data node 215. In an exemplary implementation, session information includes the network address(es) for a requested system node 230 and the identity of the client 220 authorized to access the system node 230. Session information also includes one or more conditions that the client 220 must satisfy before being authenticated by the data node 215. For purposes of illustration, the client 220 may be required to present a valid UserID and password, although other implementations are also contemplated as being within the scope of the invention (e.g., the use of security certificates or security keys).

[0031] Session information may also include other information about the client session. By way of example, session information may also include an expiration time for the client session. Upon expiration, the client 220 may no longer be able

to access the system node 230 without being re-authenticated by the control node 210. As another example, session information may identify client permissions (e.g., functions that the client 220 is authorized to access at the system host 230). Still other implementations are also contemplated, as will be readily apparent to those skilled in the art after having become familiar with the teachings of the present invention.

[0032] Authorization module 211 may also register system nodes 230 at the control node 210. During a registration operation, the system node(s) 230 provide their network address to the control node 210. Control node 210 maintains the network address in a client database 212. In an implementation using dynamic network addresses, client database 212 is updated in response to a different network address being assigned to the system node 230, or on some other recurring or periodic basis (e.g., every 4 hours).

[0033] Control node 210 may be communicatively coupled to the data node 215 (e.g., via network 200 or other suitable connection). In an exemplary implementation, data node 215 includes a session module 216 which cooperates with control node 210 to establish a connection between the client 220 and the system node 230. Session module 216 may also be implemented as computer readable program code (e.g., software, firmware) stored in computer readable storage or memory and executable by a processor (or processing units) operatively associated with the data node 215.

[0034] Session module 216 is operatively associated with a session database 217. Session module 216 populates session database 217 with session information received from the control node 210 for a client session. When the client 220 requests access to the system node 230, data node 215 uses the session information in session database 217 to establish a secure authenticated connection between the client 220 and the system node 230.

Exemplary Operations

[0035] Figs. 3a through 3f illustrate exemplary methods for implementing remote access to a system node (e.g., for a building automation system) via a secure authenticated network connection. The methods described herein may be embodied as logic instructions. When executed on a processor (or processing devices), the logic instructions cause a general purpose computing device to be programmed as a special-purpose machine that implements the described methods. In the following exemplary operations, the components and connections depicted in the figures may be used to implement a secure authenticated network connection.

[0036] In Fig. 3a, one or more system nodes 300 register with a control node 310 via a suitable communications link 301 (e.g., TCP/IP). The control node 310 authenticates each system node 300, e.g., based on information about the system node. Registration information 302 (e.g., data node and corresponding network

address) for each registered system node 300 may also be maintained in the client database 320. Other information, such as the status of a system node 300 may also be maintained in the client database 320 (e.g., online, busy).

[0037] In Fig. 3b, client 330 initiates a client session with the system node 300 by establishing a communications link 331 with the control node 310 (e.g., via HTTPS at a secure web site). The client provides authentication information 332 (e.g., UserID and password) to the control node 310. The control node 310 authenticates the client 330, e.g., based on information maintained in client database 320, and returns a data structure (e.g., list 333) identifying registered system nodes 300 that the client 330 has permission to access. The list 333 may also indicate whether the system node 300 is registered (e.g., whether the dynamic address has been updated) and the status of the system node 300.

[0038] Before continuing, it should be noted that control node 310 resides at a "known" network address (e.g., a static IP address). Accordingly, the control node 310 may be readily accessed by the system node(s) 300 (e.g., during registration) and by the client(s) 330.

[0039] In Fig. 3c, the client 330 sends a request 334 to the control node 310 identifying a registered system node from the list 333. The control node 310 verifies that the client 330 satisfies the access permissions for the requested system node 300 (e.g., based on information maintained in client database 320), and that the system node 300 is registered and available.

[0040] If the client 330 has access permissions to the requested system node 300, and the requested system node 300 is registered and available, the control node 310 generates session information 312. The control node 310 sends the session information 312 to data node 340 over communications link 311 (e.g., via a secure socket connection where it is stored in session database 350). In an exemplary implementation, the control node 310 and data node 340 may be located physically close to one another and a secure connection may be established behind a local firewall. Optionally, the control node 310 may be authenticated by the data node 340.

[0041] In Fig. 3d, a secure communications link (e.g., HTTPS) 305 is established between the control node 310 and the system node 300. The control node 310 then provides session information 306 to the system node 300. The session information 306 provided to the system node 300 may include a TCP/IP address/port/security key, and session ID for establishing connections with the data node 340.

[0042] The control node 310 also provides session information 335 to the client 330. The session information 335 provided to the client 330 may also include TCP/IP address/port/security key, and session ID for establishing a connection with the data node 340.

[0043] In Fig. 3e, the system node 300 establishes a secure communications link 341 with the data node 340 (e.g., HTTPs) and gives the data node 340 a

request for a session 342. The client 330 establishes a secure communications link 360 with the data node 340 (e.g., via a secure socket connection), and sends a request 345 for a client session with the system node 300. The data node 340 authenticates the request 345, for example, based on the session information 312 received in Fig. 3c. The client 330 is then linked to the system node 300 over a secure authenticated connection via the data node, as illustrated below with reference to Fig. 3f.

[0044] In an exemplary implementation illustrated in Fig. 3f, the client 330 may request data from the system node 300 via secure authenticated connection 360 to the data node 340. The data node 340 in turn notifies the system node 300 of the client request (e.g., via a non-secure socket 361). The system node 300 establishes a secure (optionally temporary) connection 362 with the data node 340 and returns the requested data to the data node 340 over connection 362. Data node 340 in turn returns the requested data to the client 330 over secure authenticated connection 360.

[0045] In another exemplary implementation also illustrated in Fig. 3f, the client 330 may submit a message with a command for the system node 300 via secure authenticated connection 360 to the data node 340. The data node 340 notifies the system node 300 that the message is pending (e.g., via a non-secure socket 361). The system node 300 establishes a secure (optionally temporary)

connection 362 with the data node 340 and retrieves the message from the data node 340 via connection 362. System node 300 may then execute the command.

[0046] In another exemplary implementation also illustrated in Fig. 3f, the client 330 may submit a message with configuration data for the system node 300 via secure authenticated connection 360 to the data node 340. The data node 340 notifies the system node 300 that the message is pending (e.g., via a non-secure socket 361). The system node 300 establishes a secure (optionally temporary) connection 362 with the data node 340 and retrieves the message from data node 340 via connection 362. The system node 300 may then apply the configuration data to the building automation system.

[0047] In another exemplary implementation, again illustrated in Fig. 3f, the client 330 may terminate the client session with the system node 300. The client 330 notifies the data node 340 to terminate the session via secure authenticated connection 360. The data node 340 closes all communications links (e.g., secure optionally temporary link 362 and non-secure link 361) with the system node 300. Optionally the data node 340 removes the session information for the terminated session from the session database 350.

[0048] It is noted that the connections 360, 361, and 362 may be established and re-established, or may be maintained throughout a common client session. It is also noted that the system node 300 may send status messages 370 to the control node 310 indicating its status (e.g., available, busy).

Exemplary Computing Device

[0049] Fig. 4 depicts an exemplary general purpose computer 400 capable of executing a program product and establishing a secure authenticated network connection. In such a system, data and program files may be input to the computer, including without limitation by removable or non-removable storage media or a data signal propagated on a carrier wave (e.g., data packets over a network). The computer 400 may be a conventional computer, a distributed computer, or any other type of computing device.

[0050] The computer 400 can read data and program files, and execute the programs and access the data stored in the files. Some of the elements of an exemplary general purpose computer are shown in Fig. 4, including a processor 401 having an input/output (I/O) section 402, at least one processing unit 403 (e.g., a microprocessor or microcontroller), and a memory section 404. The memory section 404 may also be referred to as simply memory, and may include without limitation read only memory (ROM) and random access memory (RAM).

[0051] A basic input/output system (BIOS), containing the basic routines that help to transfer information between elements within the computer 400, such as during start-up, may be stored in memory 404. The described computer program product may optionally be implemented in software modules loaded in memory 404 and/or stored on a configured CD-ROM 405 or other storage unit 406, thereby

transforming the computer system in Fig. 4 to a special purpose machine for implementing the described system.

[0052] The I/O section 402 is optionally connected to keyboard 407, display unit 408, disk storage unit 406, and disk drive unit 409, typically by means of a system or peripheral bus (not shown), although it is not limited to these devices. The system bus may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures.

[0053] Typically the disk drive unit 409 is a CD-ROM drive unit capable of reading the CD-ROM medium 405, which typically contains programs 410 and data. Computer program products containing mechanisms to effectuate the systems and methods in accordance with the present invention may reside in the memory section 404, on a disk storage unit 406, or on the CD-ROM medium 405 of such a system. Alternatively, disk drive unit 409 may be replaced or supplemented by a floppy drive unit, a tape drive unit, or other storage medium drive unit. The network adapter 411 is capable of connecting the computer system to a network 412. In accordance with the present invention, software instructions directed toward accepting and relaying access information (e.g., authentication and security data) may be executed by CPU 403, and databases may be stored on disk storage unit 406, disk drive unit 409 or other storage medium units coupled to the system.

[0054] The drives and their associated computer-readable media provide nonvolatile storage of computer-readable instructions, data structures, program modules and other data for the computer 400. It should be appreciated by those skilled in the art that any type of computer-readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories (RAMs), read only memories (ROMs), and the like, may be used in the exemplary operating environment.

[0055] The computer 400 may operate in a networked environment using logical connections to one or more remote computers. These logical connections are achieved by a communication device 411 (e.g., such as a network adapter or modem) coupled to or incorporated as a part of the computer 400. Of course the described system is not limited to a particular type of communications device. Exemplary logical connections include without limitation a local-area network (LAN) and a wide-area network (WAN). Such networking environments are commonplace in office networks, enterprise-wide computer networks, intranets and the Internal, which are all exemplary types of networks.

[0056] In addition to the specific implementations explicitly set forth herein, other aspects and implementations will be apparent to those skilled in the art from consideration of the specification disclosed herein. It is intended that the

specification and illustrated implementations be considered as examples only, with a true scope and spirit of the following claims.